

Magda's notes about papers read in **2022**

Last update: February 6, 2022

Informal notes for my future self who is likely to forget. I explain the papers the way I understand them, using terminology and logic natural to me. This means I may deviate from the original paper structure, notation, etc. At places, my interpretation may be incorrect due to lack of understanding. I will strive for this not to happen too often but I'm certainly not infallible.

This is a working document, not polished, with possible typos, editing errors, etc.

Imagery is usually taken directly from the original papers (unless otherwise stated).

Contents

1	graphNVP	2
2	MPGVAE	4
	Index	7

1 graphNVP

Paper: Kaushalya Madhawa et al. "GraphNVP: an Invertible Flow-based Model for Generating Molecular Graphs". 2019

ICLR not accept reviews

Official code

Local pdf

Notes taken: 6/2/2022

TLDR: Generate graphs $G = (A, X)$ through invertible flows. Dequantize A and X by uniform noise to make modelling continuous. Coupling strategy splitting the A and X matrices always by one node vs all the rest (breaks permutation invariance). The scale and translation in the coupling are graph NNs (do not have to be invertible) - for X functions of all other nodes and complete A matrix; for A edges between other nodes. Final reconstructed / generated A, X are quantized (floored) to get back to discrete.

Notes:

- Dequantize by simple uniform noise to get rid of the discreteness problem, then simply quantize (floor) to get back.
- ZINC-250K random sample of 250k molecules from complete ZINC to make experiments manageable.
- Good list of baseline models for experiments
- No validation checks during generation \rightarrow versatile and fast.
- Coupling structure by nodes breaks permutation invariance - is this really a problem?

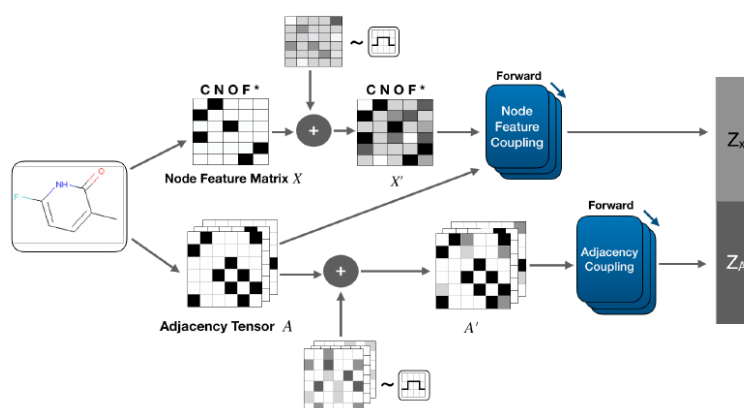


Figure 1: For the forward transform, A and X matrices are perturbed by uniform noise. The node coupling is conditioned on the A matrix.

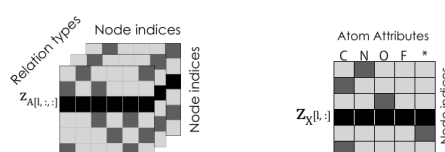


Figure 2: The coupling structure splits the A and X matrices as one node vs all the rest. This works best in experiments, though breaks down permutation invariance.

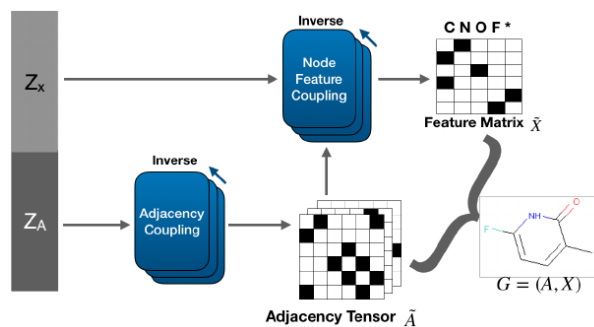


Figure 3: Generation is in two steps because the reverse coupling transform of X is conditioned on A so this needs to be generated first. Using simple dequantization (flooring) to get from continuous flow output to discrete A , X .

Experiments: on QM9 and ZINC-250K with kekulized graphs (without hydrogens) - our *cheat* version. Have not so great validity but high uniqueness. Claim benefits in speed of generation as opposed models that have complex decoders checking the validity of molecules in the process. Some comments on smoothness of latent space and possibility to interpolate for property control but not very convincing.

2 MPGVAE

Paper: Daniel Flam-Shepherd et al. "MPGVAE: improved generation of small organic molecules using message passing neural nets". In: *Machine Learning: Science and Technology* 2.4 (2021). Publisher: IOP Publishing, p. 045010

no reviews

no official code

Local pdf

Notes taken: 6/2/2022

TLDR: Builds on GraphVAE (Simonovsky and Komodakis, 2018) but bypasses the graph matching problem by relaying on message passing NN in encoder and decoder so that the reconstruction loss can be calculated directly. Message passing over edges, nodes use attention to aggregate these. To get from node representation to latent vector they use set2vec from Vinyals 2015 to be invariant to the node ordering. To get from latent back to graph they use simply linear layer and then message passing, finishing with softmax over node and edge representation to create stochastic edge and node graph representations from which the graphs can be samples (in practice assume by max category). The reconstruction loss assumes independence of node and edge features $p\theta(G|z) = \prod_{v \in G} p(x_v|z) \prod_{u \in G} p(e_{uv}|z)$

Notes:

- code not publicly available - asked first author, no answer
- set2vec to be invariant to node ordering when getting from the node representation in the last MPNN layer the latent representation z , on the way back can use MLP and MPNN cause the ordering problem was solved when moving to the latent
- *I still do not see how the reconstruction loss can be evaluated. Even with the independence assumptions, each of the node/ edge likelihoods has to be matched to the respective edge/ node in the original graph. How can this be done without some matching procedure?*

Graph with adjacency matrix, edge feature tensor and node feature matrix $G = (A, E, X)$. Assume independence between node and edge probability distributions (Erdos-Rényi graph model) such as in "GraphVAE" of Simonovsky and Komodakis (2018).

Starting point is standard VAE and ELBO. Both edge and node features are categorical. Build a message passing NN (MPNN) into encoder and decoder of the VAE.

Refers to Graphite from Grover et al 2018 as nearest baseline - how different is this?

Claim that has lower complexity than GraphVAE (how so?).

Encoder: Use MPNN to map node and edge representations into latent space. Messages are constructed at edge level as linear functions of itself, neighbouring nodes, and edge features passed through tanh nonlinearity. Edge updates are directly the messages. Node updates first use attention to reweight the messages and then pass it through a GRUCell (this is not very clear, there is something strange in eq. 11 - should be a message for u only? not uv ?) Use set2set from Vinyals 2015 (unlike in seq2seq, in sets the order should not matter and the model should produce the same representation in arbitrary ordering of the set) to read from last node representation to fixed-sized hidden representation z .

Decoder: Pass latent through linear layer to expand dims and then RNN to get initial node features (edge features zero - unconnected graph). MPNN identical to encoder - update both nodes and edges. Readout the last layer into stochastic graph representation by passing it through single layer NN and softmax.

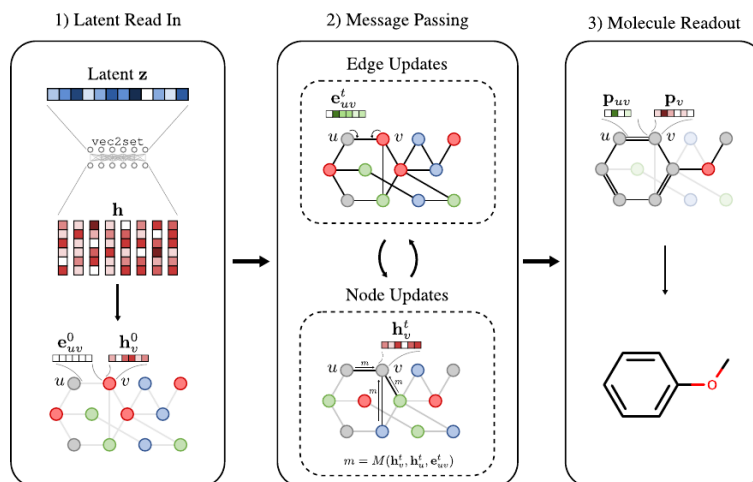


Figure 4: 1) Read in latent z by passing it through linear layer to get to correct node dimensions and pass it through RNN to get initial node features. Edge features fixed to zero - unconnected graph. 2) Message passing on both, node and edge representations. 3) Use last node and edge representation to predict independent node and edge feature probabilities.

Experiments: Over QM9 (not clear whether kekulized or not and if has hydrogens) - best validity compared to GrphVAE, MolGAN, CharacterVAE and Grammar VAE. Generate 10^4 molecules and achieve validity over 90%, unique almost 70% and novel about 50% - much better than baseline. Also evaluates the continuous (chemical properties) and discrete (avg number of atom types and ring size) property distributions of generated vs training sample graphs - claim to have both better than baseline.

References

- [1] Daniel Flam-Shepherd, Tony C. Wu, and Alan Aspuru-Guzik. “MPGVAE: improved generation of small organic molecules using message passing neural nets”. In: *Machine Learning: Science and Technology* 2.4 (2021). Publisher: IOP Publishing, p. 045010 (cit. on p. 4).
- [2] Kaushalya Madhawa et al. “GraphNVP: an Invertible Flow-based Model for Generating Molecular Graphs”. 2019 (cit. on p. 2).
- [3] Martin Simonovsky and Nikos Komodakis. “GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders”. In: *arXiv:1802.03480 [cs]* (2018) (cit. on p. 4).

Index

dequantize, 2

February 2022, 2, 4

graph, 2

graphNVP, 2

GraphVAE, 4

invertible flow, 2

message passing NN, 4

MPGVAE, 4

QM9, 3

set2vec, 4

ZINC, 3